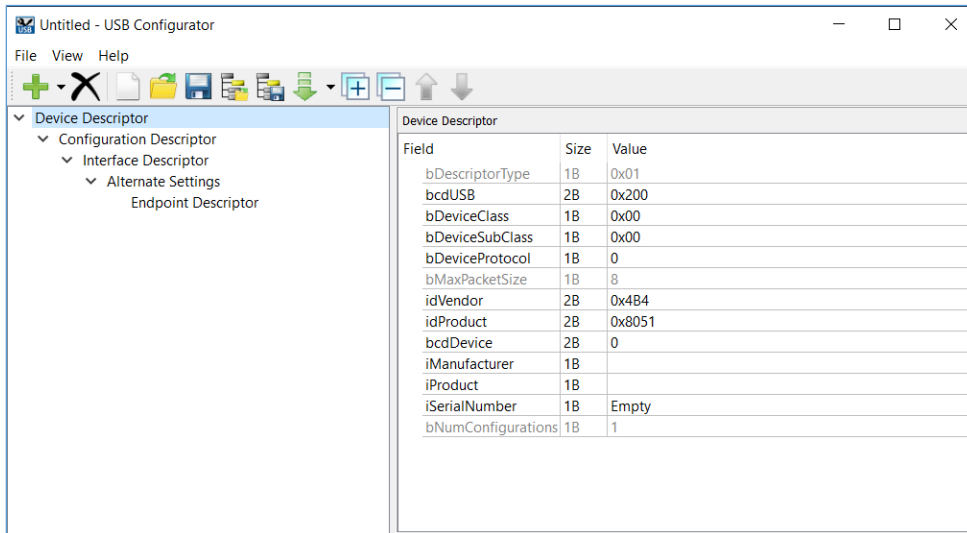


## Overview

The USB Configurator is a stand-alone tool included in the ModusToolbox software to configure USB Device descriptors. See the [Supported Descriptors](#) section for a list of supported USB descriptors. After configuring and saving a USB Device descriptor, the USB Configurator generates configuration files that store USB Device descriptors and other information (see [Code Generation](#)) used by the USB Device middleware configuration and operation.



## Supported Middleware

Name	Link
Cypress PSoC 6 USB Device Middleware Library	<a href="https://github.com/cypresssemiconductorco/usbdev">https://github.com/cypresssemiconductorco/usbdev</a>

## Launch the USB Configurator

You can launch the USB Configurator as a GUI with or without the ModusToolbox IDE. You can also run the tool from the command line. The USB Configurator GUI contains [menus](#), [toolbars](#), and two [panes](#) used to configure device descriptors. The command line tool has various options.

### Launch without the ModusToolbox IDE

To launch the USB Configurator GUI without the ModusToolbox IDE, navigate to the install location and run the executable. On Windows, the default install location for the USB Configurator is:

```
<install_dir>/tools_<version>/usbdev-configurator
```

For other operating systems, the installation directory will vary, based on how the software was installed.

The USB Configurator opens with an untitled configuration file (\*.cyusbdev). Save it as a new file and provide a file name, or open another existing \*.cyusbdev file.

## Launch with the ModusToolbox IDE

If there is a \*.cyusbdev file in the application folder, you can launch the USB Configurator GUI directly from the ModusToolbox IDE using any of the following methods:

- Double-click on the \*.cyusbdev file in the application.
- Right-click on the top-level application folder, and select **ModusToolbox > USB Configurator**.
- Click on the "USB Configurator" link in the Quick Panel, under **Configurators**.

If there is no \*.cyusbdev file in the application folder, the options from the menu and Quick Panel read **USB Configurator (new configuration)**. Select either option, and the USB Configurator opens with a default configuration (\*.cyusbdev) that will be saved to the *design.cyusbdev* file in the application folder.

## From the Command Line

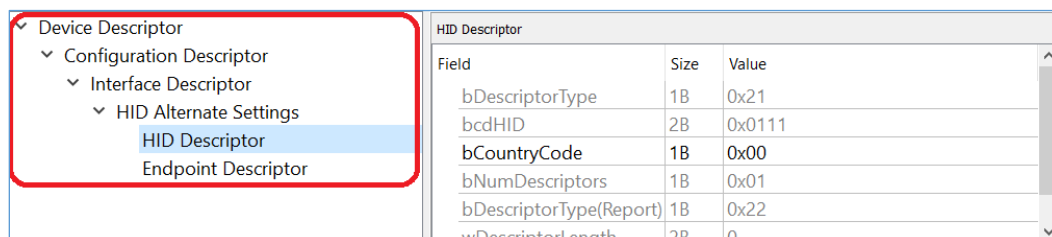
You can run the usbdev-configurator executable from the command line. However, there are only a few reasons to do this in practice. There is also a usbdev-configurator-cli executable, which re-generates source code based on the latest configuration settings from a command-line prompt or from within batch files or shell scripts. The exit code for the usbdev-configurator-cli executable is zero if the operation is successful, or non-zero if the operation encounters an error. In order to use the usbdev-configurator-cli executable, you must provide at least the --config argument with a path to the configuration file.

For more information about command-line options, run the usbdev-configurator or usbdev-configurator-cli executable using the -h option.

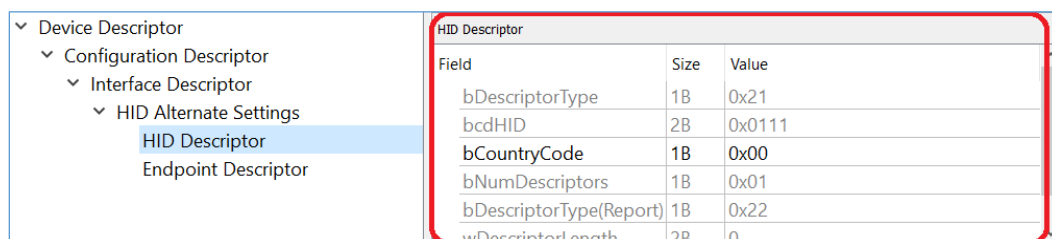
## Quick Start

This section provides a simple workflow for how to use the USB Configurator.

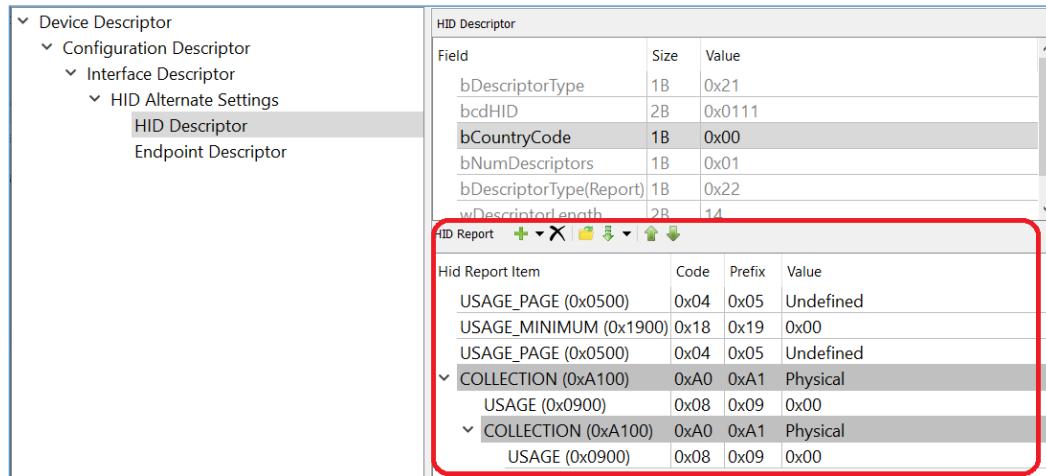
1. [Launch the USB Configurator](#).
2. Configure the device descriptors hierarchy in the **Device Descriptor** pane. See the [Descriptors](#) section.



3. Configure device descriptor parameters in the Parameter pane.



- The Parameter pane contains a sub-pane for HID descriptor (HID report pane).



- Save the configuration.  
See [Code Generation](#).
- Use the generated structures as input parameters for functions in your application.

## Code Generation

The USB Configurator generates header (.h) and source (.c) files that contain relevant firmware used by the USB Device middleware configuration and operation. The firmware contains arrays to store USB Device descriptors, structures that help the middleware to access descriptors, middleware and classes configuration structures, and a set of defines. The generated files *cycfg\_usbdev.h* and *cycfg\_usbdev.c* are located in the *GeneratedSource* folder next to the \*.*cyusbdev* file.

Refer to the USB Device Middleware Library for more information about this code.

## Menus

- **File**
  - New** – Creates a new file with new configuration.
  - Open** – Opens the configuration file.
  - Save** – Saves the existing file.
  - Save As** – Saves the existing file under a different name.
  - Load Descriptor** – Loads a descriptor.
  - Save Descriptor** – Saves a descriptor.
  - Exit** – Closes the configurator.
- **View**
  - Notice List** – Shows/hides Notice List. See *Device Configurator Guide* for more information about Notice List.
  - Toolbar** – Shows/hides the toolbar.
  - Reset View** – Resets the view to the default.

- **Help**

- **View Help** – Opens this document.
- **About USB Configurator** – Opens the About box for version information.

## Toolbars

### Descriptor Toolbar

Provides basic commands to open and save files and to configure the descriptors hierarchy.



- **Add descriptor** – Adds a new sub-descriptor or Device descriptor.
- **Delete descriptor** – Deletes the selected descriptor.
- **New** – See [Menus](#).
- **Open (Ctrl+O)** – See [Menus](#).
- **Save (Ctrl+S)** – See [Menus](#).
- **Load Descriptor** – See [Menus](#).
- **Save Descriptor** – See [Menus](#).
- **Import Descriptor** – Selects a descriptor from the pull-down menu to import.
- **Expand all tree items** – Expands all items in the descriptor tree.
- **Collapse all tree items** – Collapses all items in the descriptor tree.
- **Move Up** – Moves up the selected descriptor.
- **Move Down** – Moves down the selected descriptor.

### HID Report Toolbar

Provides the commands to configure an HID descriptor report.



- **Add HID report item** – Creates a new HID report item for the current HID Descriptor.
- **Delete HID report item** – Deletes the selected HID report item for the current HID Descriptor.
- **Load HID report** – Loads an HID report file generated by the current HID Descriptor.
- **Import HID report** – Imports an HID report item for the current HID Descriptor.
- **Move up HID report item** – Moves up an HID report item.
- **Move down HID report item** – Moves down an HID report item.

## Panes

The USB Configurator contains two panes that display information about descriptors and their parameters:

### Descriptors

This pane shows the descriptors hierarchy.

**Note** All descriptors have their own hierarchy that can be created when their parent is selected. However, the **Device Descriptor** is a root descriptor and it has no parent. The **Device Descriptor** can be created from any descriptor and will be added to the end of the tree.

To add specific descriptors such as **CDC Interface Descriptor** or **HID Descriptor**, add a special parent descriptor:

- Add CDC Interface Descriptor for CDC Descriptor
- Add HID Alternate Settings for HID Descriptor.

### Parameters

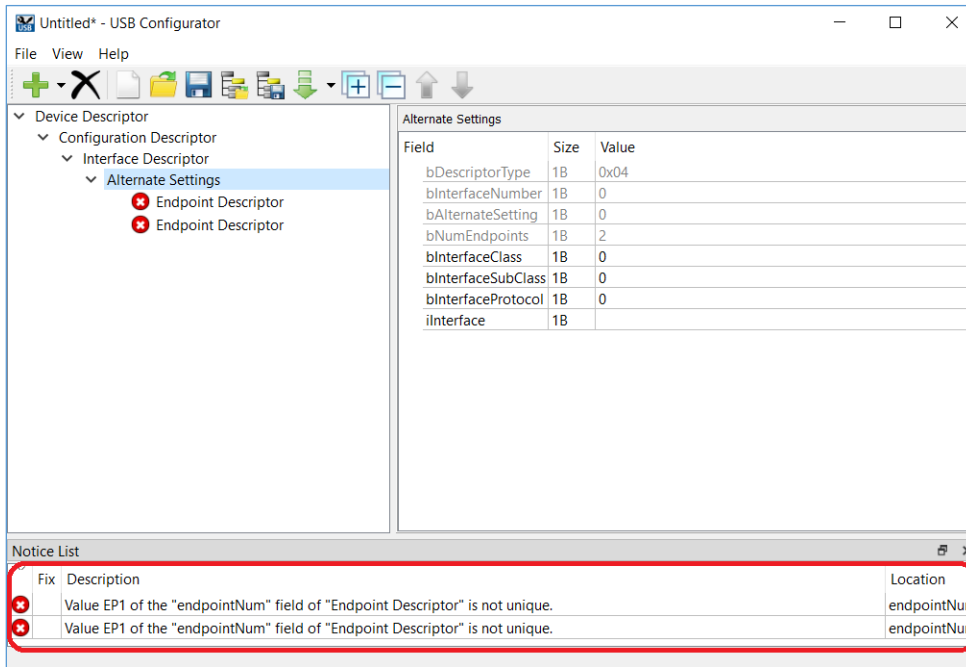
This pane shows configuration information for the selected descriptor.

**Note** The Parameter pane has different controls to edit different parameters (text box, combo box, or multi-line text box). Most parameters have a text box as the editing control.

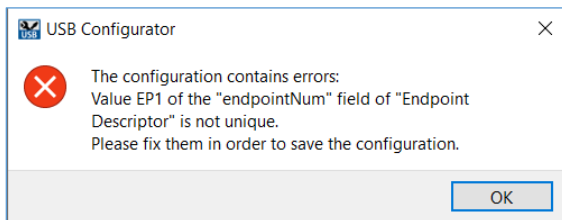
- **Vendor-defined items** – Some string parameters or **HID report items**, such as **iSerialNumber**, have a combo box with a list of predefined items. The “Empty” value is selected by default. To specify a value not on the list, select “Vendor-defined.” The combo box will change to a text box to type an appropriate value. To return to the combo box, erase the value and leave the control.
- **String pool** – Parameters such as **iChannelNames** in **AC Processing Unit** of **Audio Interface Descriptor 1.0** support a string pool and require a multi-line text box. To insert several strings, use an end-line separator.
- **Read-only parameters** – There are two types of read-only parameters: predefined **bDescriptorType** or auto-calculated **bConfigurationValue**.
- **Array parameters** – Parameters such as **bSubordinateInterface** in union with **Communication Alternate Settings** is an array. Use “;” to separate elements.
- **Hexadecimal/Decimal** – When a value starts with “0x,” it is parsed as a hexadecimal value. In other cases, it is parsed as a decimal.
- **Map and bit fields** – Some parameters, such as **bmAttributes**, are read-only by themselves. However, you can insert them using the related bit fields below them in the parameters list. **Bit fields** have a size of 0B. Their name starts with the name of a field whose part, plus the bits for which they are responsible. For example, **bmAttributes(1-0): Transfer Type**.

## Errors

The USB Configurator has a validation system that can identify errors with icons and an error message in a tooltip, as well as in Notice List.



When you try to save the configuration with errors, a message shows the problem to fix.



## Supported Descriptors

All supported descriptors are described by USB Implementers Forum, Inc. You can find more details at <http://www.usb.org>.

The list of supported descriptors:

- Device, Configuration, IAD (Interface Association Descriptor), Interface, Endpoint
- Audio v1.0 and v2.0 descriptors
- CDC Communication descriptors:
  - Header Functional descriptor
  - Union Functional descriptor
  - Country Selection Functional descriptor
  - Call Management Functional descriptor (PSTN)
  - Abstract Control Management Functional descriptor (PSTN).

- HID descriptor and HID Report descriptor
- BOS descriptors (including Container ID and USB 2.0 Extension descriptors)
- Microsoft OS descriptors v1.0

**Note** An Interface Descriptor is represented by two items to build a tree structure: Interface Descriptor with an empty parameters pane and Alternate Settings with all Interface Descriptor parameters.

**Note** Class-Specific AS Encoder/Decoder Descriptors from USB Audio v2.0 are not supported.

## Known Issues, Limitations, and Workarounds

The USB Configurator supports import from the HID Descriptor Tool. Current version 2.4 contains an error related to strings. Per spec HID1.11 String items should have such values:

- String Index 0111 10 nn
- String Minimum 1000 10 nn
- String Maximum 1001 10 nn

But the HID Descriptor Tool generates:

- String Index 0110 10 nn
- String Minimum 0111 10 nn
- String Maximum 1000 10 nn

Before importing, fix these items manually in the file generated with the HID Descriptor tool.

There are [several possible flows](#) to run the USB Configurator. To migrate configurations between flows, you must regenerate code for each flow.

## Migration of Configuration File Format

Versions of the USB Configurator prior to 2.0 used a C header file to store its configuration as a comment in XML format. In version 2.0, the configuration is stored in a separate `.cyusbdev` file in XML format. Use the following instructions to migrate to the `.cyusbdev` file as appropriate:

1. Launch the USB Configurator.
2. Click **Open**.
3. Select the **Obsolete configurator files (\*.h)** file extension and choose a header file.
4. After the configuration is loaded, click **Save** to create the `.cyusbdev` file and update the C file.

### Notes

- The `*.cyusbdev` file is located one directory up related to the header file.
- The command-line argument `--config` does not accept obsolete configuration files (\*.h).

## References

Refer to the following documents for more information, as needed:

- ModusToolbox IDE User Guide
- Cypress USBFS Device Middleware Library Documentation
- <http://www.usb.org>

## Version Changes

This section lists and describes the changes for each version of this tool.

Version	Change Descriptions
1.0	New tool.
1.1	Updated the icons to be standard. Added Notice List.
2.0	Changed the user configuration storage location from the header file to the *.cyusbdev file.
	Added <b>New</b> , <b>Save As</b> , <b>Reset View</b> commands. Changed the <b>Load</b> command to <b>Open</b> .
	Updated the icons.
	Removed the <b>CUSTOM</b> item from <b>HID Report</b> .
	Removed the functionality to launch the USB Configurator from the Device Configurator.

© Cypress Semiconductor Corporation, 2018-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries, including Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited. Spansion LLC ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. Cypress products are not designed, intended, or authorized for use as critical components in systems designed or intended for the operation of weapons, weapons systems, nuclear installations, life-support devices or systems, other medical devices or systems (including resuscitation equipment and surgical implants), pollution control or hazardous substances management, or other uses where the failure of the device or system could cause personal injury, death, or property damage ("Unintended Uses"). A critical component is any component of a device or system whose failure to perform can be reasonably expected to cause the failure of the device or system, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from or related to all Unintended Uses of Cypress products. You shall indemnify and hold Cypress harmless from and against all claims, costs, damages, and other liabilities, including claims for personal injury or death, arising from or related to any Unintended Uses of Cypress products.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners. Spansion logo, and combinations thereof, ModusToolbox, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit [cypress.com](http://cypress.com). Other names and brands may be claimed as property of their respective owners.